

A Semi-Automatic Light-Weight Approach Towards Data Generation for a Domain-Specific FAQ Chatbot Using Human-in-the-Loop

Anum Afzal^a, Tao Xiang^b and Florian Matthes^c

School of Computation, Information and Technology, Technical University of Munich, Boltzmannstrasse 3, 85748, Garching bei Muenchen, Germany

Keywords: FAQ Chatbot, Efficient Transformer, Training Data, Natural Language Generation, NLG Evaluation.

Abstract: Employees at large companies tend to have longer waiting times if they need company-specific information and similarly someone on the other end needs to manually address those queries. Most companies are trying to incorporate LLM-powered conversational agents to make this processing faster but often struggle to find appropriate training data, especially domain-specific data. This paper introduces a semi-automatic approach for generating domain-specific training data while leveraging a domain-expert as a human-in-the-loop for quality control. We test this approach on a HR use-case of a large organization through a retrieval-based question-answering pipeline. Additionally, we also test the effect of long context on the performance of the FAQ chat for which we employ LongT5, an Efficient Transformer. Our experiments using LongT5 show that the inclusion of the generated training data improves the performance of the FAQ chatbot during inference.

1 INTRODUCTION


The rapid advancement of artificial intelligence (AI) and natural language processing (NLP) techniques has led to a growing interest in developing intelligent chatbots for a wide range of applications (Zaib et al., 2021). Naturally, many large organizations are adopting these conversational agents to automate the information retrieval processes and hence, making them faster and more convenient (Nicolescu and Tudorache, 2022). One such application is the human resource (HR) domain, where chatbots can help answer employee queries, provide information about company policies, and assist with various HR-related tasks.


However, one challenge in designing such a conversational agent is that users are often unpredictable and would often deviate from the training phrases used during model training. This includes asking ambiguous or incomplete questions, making it difficult for the chatbot to address them properly. This becomes even more critical when a Language Model (LM) powered Retrieval Augmented Pipeline (RAG) is used. Such an approach first finds the relevant article to be used as grounding knowledge, which together with the user


question is passed to a Language Model (LM), that generates the answer addressing the user query. If the user utterance is matched to the wrong article then the probability of the model generating the correct answer is automatically quite low. Similarly, an Instruct-tuning (Wei et al., 2022) could never reach its full potential if the instructs deviate too much from the real-world distribution.

To address this challenge and ensure the chatbot's relevance and effectiveness, we present and evaluate a semi-automatic approach for generating training data following the real-world distribution. While we implement and evaluate it for the HR domain of a large organization, this can be applied to many other domains. We deployed an HR chatbot powered by a question matching algorithm, and used the logs to construct an additional training set consisting of real user utterances. We also introduce the concept of the human-in-the-loop who is a domain experts curating the quality of the generated dataset.

When working with a retrieval-based approach, another challenge in passing grounding knowledge to the chatbot is the document length, which typically exceeds the 512 token limitations of most transformer-based architectures. To address this issue, we investigate the use of Efficient Transformers (Tay et al., 2022), specifically the LongT5 model (Guo et al., 2021), which is designed to handle longer sequences more

^a  <https://orcid.org/0000-0001-8146-1949>

^b  <https://orcid.org/0000-0001-6217-6560>

^c  <https://orcid.org/0000-0002-6667-5452>

effectively than its predecessor, the T5 (Raffel et al., 2019).

To summarize, we address the following two challenges in this paper:

- When interacting with a chatbot, users may stray from the phrases and queries used during the training phase, which can effect chatbot performance. We address this by collecting training phrases from the chatbot logs and using it for model training to make it more robust during inference.
- We highlight the limitation of transformer-based language models in terms of the context window and advocate the usability of an Efficient Transfer supporting a larger context window. Additionally, we also evaluate the model’s performance across various context windows.

Since the use case revolves around frequently asked questions, we have limited the scope to single-turn conversations. Furthermore, to test the effectiveness of the generated data, we fine-tune the LongT5 model with and without the inclusion of the training data collected via chatbot log, allowing us to compare the performance under both data distributions.

2 RELATED WORK

The development of conversational agents has been an active area of research in recent years. With the recent advancements in Natural Language Processing, there has been a massive surge in the number and capabilities of Large Language Models (LLMs). For example, GPT-3 (Brown et al., 2020) and its 175B parameter count enable it to model language in a similar degree to that of a human. Building on these advancements and coming closer to the chatbot use-case, we refer to the question generation approach where LLMs could be leveraged to generate sample questions given an article ((Ushio et al., 2023; Leite and Lopes, 2023)), but the generated questions are still structured and still far away from the actual user utterances that a chatbot encounters during run-time.

Many state-of-the-art LLMs can now be Instruct-tuned to make them follow instructions. While user queries can oftentimes be ambiguous, the Instruction-tuning approach could be used to make the chatbot more receptive towards them and accurately address them. In a similar direction, albeit with a markedly different approach, Instruct-GPT (Ouyang et al., 2022) utilizes the reinforcement learning PPO technique (Schulman et al., 2017): following an initial instruction fine-tuning procedure on data collected through

the OpenAI API¹, a reward model was constructed to reflect human preferences on different instructions following model outputs. With these components, researchers were then able to further train the baseline through policy optimization, outperforming both the base fine-tuned model and a FLAN-fine-tuned version. A similar approach could be utilized by many large organizations when designing their internal FAQ chatbots but instead of relying on OpenAI for instruct-tuning data, companies could generate their data for model tuning.

3 BACKGROUND

This section will provide a refresher on the concepts that serve as an essential building block of the framework presented in this paper.

3.1 Dense Passage Retriever

A Dense Passage Retriever (DPR) typically refers to a module that has been, given an input (question), trained to retrieve the top-k pertinent documents from a large collection of documents. Such a system (Karpukhin et al., 2020) could use sparse vector space models such as BM-25 and TF-IDF to embedding-based approaches that utilize models like BERT. The former approach relies on the bag-of-words representation of text. Whereas, the latter approach makes use of the contextualized embedding from encoder-based models like BERT and its variants to find the top-k relevant documents. Several approaches also focus on a Graph-based approach for finding the most relevant documents (Albarede et al., 2022).

3.2 Generative Question Answering

Generative Question Answering (GQA) differs from standard question-answering in the sense that instead of finding the location of the answer in the reference text (Rajpurkar et al., 2016), it generates the answer from scratch. Traditionally, an encoder-decoder (seq2seq) model like T5 (Raffel et al., 2019) and BART (Lewis et al., 2019) that takes text as input and produces text as output is most suitable for a Generative Question Answering (GQA) task. While T5 poses each downstream task as a seq2seq task via prefix modeling, BART uses a similar pre-training objective, and unlike T5, only focuses on the seq2seq task. However, contemporary decoder-only LLMs like GPT 3.5/4 are now also able to effectively handle a GQA task. The

¹<https://platform.openai.com/>

standard practice is to pass the grounding knowledge along with the question as input to the model. The grounding knowledge is used to ensure that the model produces the factually correct answer. State-of-the-art large language models that have been pre-trained on a huge corpus allow us to obtain competitive results on a specific task by fine-tuning on a smaller dataset.

3.3 Efficient Transformers

The heart of the Transformer is the $n \times n$ self-attention matrix that provides the weights for the dependency of one word to another within a sentence of length n (Vaswani et al., 2017). While the Transformer architecture has greatly improved the performance of GQA models, it also introduced quadratic computational complexity and memory requirements. To address these limitations, several Efficient Transformer variants have been proposed (Tay et al., 2022). Having managed to eliminate the quadratic complexity of the self-attention, the models such as BIGBIRD-Pegasus (Zaheer et al., 2020), PEGASUS-X (Phang et al., 2023), BART-LS (Xiong et al., 2022) and LongT5 (Guo et al., 2021) remain competitive. These architectures have a self-attention matrix that incorporates Local Attention combined with some notion of Global Attention to effectively encode the input text.

Efficient Transformers arise as promising solutions given the reduced complexity of their self-attention matrix. LongT5 being the efficient variant of the T5 model is particularly relevant. The authors integrate attention ideas from long-input transformers (ETC) (Liu et al., 2020) and pre-training strategies from summarization pre-training (PEGASUS) (Zhang et al., 2020) into the scalable T5 architecture.

4 METHODOLOGY

In this section, we formulate the task and the proposed methodology in detail. However, an overview of the use-case is summarized below.

Given is a set of employee questions, and a set of context documents containing information relevant to the HR domain. Each question is associated with a context that provides the necessary information to answer the question. We opted for a retrieval augmented approach that first uses a Dense Passage Retriever trained to retrieve the context document for a given question. Secondly, a generative question-answering model that, given a question and its corresponding context document, can generate a response that is close to the ground truth answer and hence, addresses the user query. Figure 1 shows the overall architecture

of the proposed methodology, and also outlines the training triplets of question, reference answer and the context document used for fine-tuning the model for this specific task.

With the problem statement and notation introduced, we now proceed with the methodology, including the description of models, Dataset Collection, Dataset Pre-processing, and Evaluation Methods.

4.1 Datasets

4.1.1 Data Collection

A question-answering task using grounding knowledge requires a dataset where each sample is a tuple of a question and its respective answer along with context as shown in Figure 1. The HR domain experts compiled a dataset of frequently asked questions along with textual context using the company’s internal documentation. Additionally, a primitive question-matching approach was embedded in a chatbot and deployed, allowing employees of a large company to interact with it. A user query is match to a FAQ in the database and the respective answer is returned to the answers. We collected the user queries along with the matched FAQ pair, and the respective article that FAQ belonged to in the form of (*user query, matched question, respected answer, respective document*). To ensure the quality, these logs were then controlled by domain experts to ensure the quality and correctness of the dataset. As per the assessment done by the domain-experts, roughly 60% of the user queries were mapped to the correct question. Thus, we constructed two similar datasets with the differences highlighted below:

1. **FAQ Dataset:** The first dataset consists of structured questions, context, and answers derived only from the internal FAQs and handcrafted by the domain experts.
2. **UT Dataset:** The second dataset comprises user utterances as questions collected during user interaction. A retrieval model matched user utterances to questions in the FAQ Dataset. The associated contexts and answers of the matched questions were then extracted and used as the paired contexts and answers for the user utterance (questions). To ensure correctness, the dataset was inspected and manually corrected by the domain experts.

4.1.2 Dataset Analysis

Since the FAQ dataset has been manually constructed by domain experts, it classifies as the gold standard dataset. The question range is fairly distributed between all of the available topics related to the HR

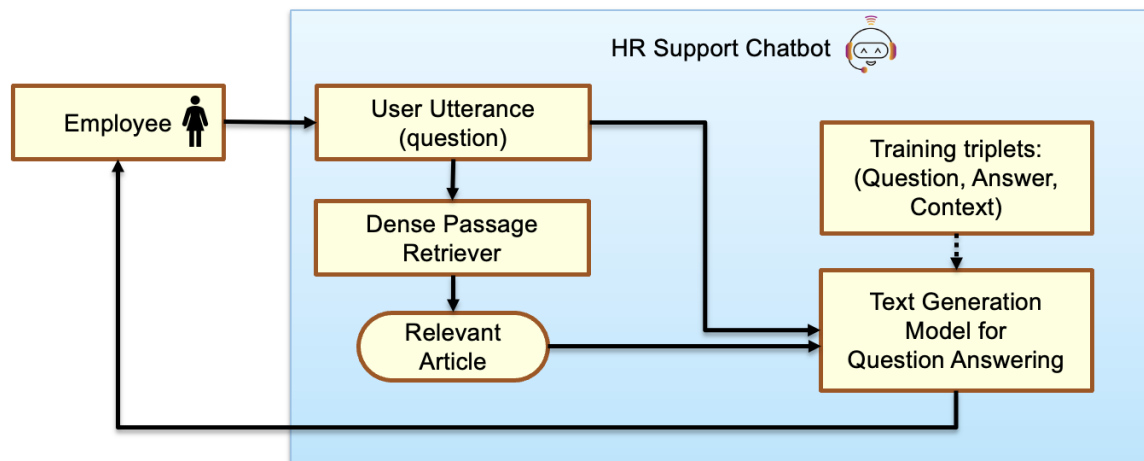


Figure 1: A block diagram depicting the flow of the Retrieval-based Question Answering approach used to evaluate the performance of the generated dataset. The dotted line represents the fine-tuning of the model while the normal line represents the system flow during inference.

domain. A small sample of such questions is shown in Figure 2 depicting questions with proper grammar and context.

Meanwhile, the questions from the UT dataset are shown in Figure 3. They are very unstructured, lacking in terms of grammar, and in some cases contain insufficient context for the chatbot to reply appropriately. Additionally, we observe the user utterance dataset to be quite skewed in terms of query distribution, containing favorite topics that are inquired about much more frequently than others.

4.1.3 Dataset Pre-Processing

In the preprocessing stage, we removed samples containing invalid values, such as NaN or purely numeric values. From here, we sample two training datasets for our experiments using the following distribution:

1. FAQ dataset ($N \approx 48k$): This dataset is derived entirely from the manually annotated frequently asked questions, containing structured question-context-answer triples.
2. FAQ + UT dataset ($N \approx 89k$): This dataset is constructed by merging the FAQ dataset and the UT dataset.

During fine-tuning, we filtered the FAQ dataset and FAQ + UT dataset with respect to the input lengths. This along with the number of samples for each filtered configuration is summarized in Table 1.

As a final step, we divided them into training, validation, and test sets, with the validation and test set each comprising 10% of the data.

Table 1: Summary of training samples used for fine-tuning after filtering the datasets for a specific input number of tokens using the T5 Tokenizer.

Datasets	Tokens		
	512	5120	7168
FAQ	4023	42914	45800
FAQ + UT	5480	77916	85977

4.2 Retrieval-Based Question Answering

4.2.1 Dense Passage Retriever

To ensure the factual correctness of the answers generated by the model, we provide the model with some grounding knowledge, which is the relevant article in this scenario. For the sake of simplicity, we used the Dense Passage Retriever (DPR) provided by Haystack². We fine-tuned a SentenceBERT model using all our HR articles, which is then used by the DPR module. We additionally used a contrastive loss function (Khosla et al., 2021) to train the DPR to assign the given user utterance to the relevant article. Specifically, we did negative mining to get the most similar but incorrect HR articles for each query based on the cosine similarity of their naïve TF-IDF embeddings. In this case, the incorrect articles refer to articles that do not contain the answer to the asked question. During inference, the DPR module embeds the user utterance and computes a similarity matrix between it and the pre-computed embedding of every document. Then we select the top-k (empirically we chose $k = 3$ to

²<https://haystack.deepset.ai/>

FAQ Question: Which departments do learners pass through by Apprenticeship ?

FAQ Question: Where can I request the Parking Space ?

FAQ Question: Can I claim back my petrol expenses? How Can I claim back my petrol expense ?

FAQ Question: Do I have to pay Australia Tax while I have different visa status ?

FAQ Question: When do I have to inform my manager about my pregnancy so I can request for my Maternity Leave ?

FAQ Question: Where can I record overtime ?

FAQ Question: I have would like to initiate an Assignment, who should I contact ?

FAQ Question: If my Religious holiday falls on weekend, can I transfer it to the following working day ?

FAQ Question: Why is there a gross salary increase on my payslip compared to last month ?

FAQ Question: How can I record a day off to go to my son's/wife's graduation in the ESS ?

Figure 2: Snippet of sample questions from the FAQ dataset.

User Utterance Question: what is my cost center

User Utterance Question: Can I register my sister for medical insurance under dependent siblings?

User Utterance Question: i want to see the compensation depending on the T level

User Utterance Question: learning

User Utterance Question: where can I find my % contribution to 401K

User Utterance Question: what is RSU-Tax and Payout

User Utterance Question: i need to update a c user extension date

User Utterance Question: transition to medicare at 65

User Utterance Question: Harmony Incentive Link

User Utterance Question: Had a doubt regarding compensation review

Figure 3: Snippet of sample questions from the UT dataset compiled using chatbot logs.

keep the sum of the document length within the input length limit of the preceding module) as the most related documents to the query.

4.2.2 Generative Question Answering Models

We experimented with and explored the performance of LongT5. We chose LongT5 as a representative of the Efficient Transformer family, specifically the local attention variant, over the transient-global (tglobal) variant for the following reasons: Firstly, the local attention-based LongT5 exhibits linear complexity concerning input sequence length, making it more efficient for processing lengthy input texts. This is achieved through the sparse sliding-window local attention operation, which allows a given token to attend to only r tokens to its left and right, with $r=127$ by default. This yields a linear complexity of $O(l \cdot r)$ where l is the input sequence length (Guo et al., 2021). Secondly, the local attention mechanism does not introduce any new parameters to the model, maintaining its simplicity. For LongT5, we employ the local attention-

based variant³, which consists of 296 million trainable parameters. This model incorporates local attention mechanisms, allowing it to efficiently process long input sequences.

4.3 Evaluation Methods

In this subsection, we outline the evaluation methods used to measure the performance of the LongT5 model trained on the two distinct datasets. To gauge the question-answering capabilities in real-world situations, we evaluate the model performance on an evaluation set derived from the UT dataset.

4.3.1 Evaluation Datasets

Evaluation Dataset ($N \approx 4k$): This dataset comprises only real user questions with contexts and answers derived from the UT dataset. We also filter the dataset to only include samples with less than or equal to (512,

³<https://huggingface.co/google/long-t5-local-base>

5120, 7168) tokens when evaluating the models with different context lengths.

4.3.2 Evaluation Metrics

To evaluate the quality of the generated responses, we use the ROUGE score (Lin, 2004) and BERTScore (Zhang et al., 2019). The ROUGE score measures the similarity between the generated response and the ground truth answer by comparing the overlap of n-grams, with higher scores indicating better performance. BERTScore, on the other hand, computes token-wise cosine similarities between the contextual embeddings of the generated and ground truth sentences, using the F1 score for aggregation. By combining these metrics, we can assess the models' performance in terms of both lexical and semantic similarities to the ground truth answers

5 EXPERIMENTS AND RESULTS

We ran experiments on LongT5 using the two dataset distributions introduced previously in the section 4.1.3. This enabled us to compare the performance of LongT5 and assess the influence of varying input lengths on the quality of the generated answers, while measuring the impact of the generated dataset on the model's performance.

To test the long-range capabilities of the LongT5 model, we tested the model by increasing the input length. We filtered and then fine-tuned on three dataset subsets, constructed by filtering out samples having more than 512 tokens, 5120 tokens, and 7168 tokens. This allowed us to include $\sim 9\%$, $\sim 90\%$, and $\sim 99\%$ of the training corpus for 512 tokens, 5120 tokens, and 7168 tokens respectively. We limited the tokens to 7168 as it allowed us to include 99% of the training corpus and the remaining 1% are very large documents which would increase the memory requirement and training time by a lot without adding a lot to the model performance.

Since the GQA model relies on the context document to generate the correct answer, the accuracy of the DPR module explained in section 4.2.1 would affect its performance. Our experiments show that we can achieve an 89.4% accuracy on the validation set with the top-1 relevant document predicted by our current DPR model. The focus of this paper was not on increasing the accuracy of the DPR, but rather on evaluating the chatbot performance having included a dataset compiled from chatbot logs for training, under the assumption that the context document passed to the GQA model is the correct one.

To test the performance of the models at run-time, we extracted a small portion of the dataset prepared using real-time user utterances as previously explained in section 4.3.1.

Based on the analysis using the ROUGE and BERTScore shown in Table 2, there are two important findings discussed below:

1. The choice of training dataset has a significant impact on the performance of the models. Our results indicate that models trained using a combination of structured FAQ pairs and our generated dataset perform better during inference time, as they more closely resemble real-world data. This is demonstrated in Table 2, where the models fine-tuned on a combination of the FAQ dataset and the UT dataset have a higher ROUGE and BERTScore than being trained on the FAQ dataset alone. However, BERTScore seems to be much higher than the ROUGE, pointing towards the unreliability of these evaluation scores.
2. While LongT5 didn't perform well on 512 tokens, it improved drastically when the context length was scaled to 5120 tokens. On the contrary, when LongT5 is scaled up to 7168 tokens, we witness a decrease in scores. This indicates that a larger context window and a larger dataset make generative question-answering a more challenging task for this specific model.

The comparison of this work with the previous baseline is difficult because the previous approach was evaluate in terms of accuracy, if the user utterance was mapped to the correct question or not. The evaluation of the text generation model in this paper is done via ROUGE & BERTScore making it challenging to compare the two approaches.

6 CONCLUSION

We investigated the performance of the collected dataset on LongT5 through GQA tasks, particularly focusing on the impact of the generated dataset. We fine-tuned LongT5 with and without the collected dataset and also varied the context length for LongT5. Our findings reveal that the choice of training dataset plays a crucial role in model performance and gets the best results when scaled up to 5120 tokens. The models that had seen the UT dataset collected via chatbot logs were able to more accurately understand user utterances during inference. Many large organizations can leverage this semi-automatic approach to generate more training data that can support training for robust FAQ chatbots in the form of Instruct-tuning or Fine-

Table 2: Evaluation of the fine-tuned LongT5 model on varying input lengths and dataset distribution during inference, both on a scale of 1 to 100, with 100 being the best.

Datasets	ROUGE			BERTScore		
	512 tokens	5120 tokens	7168 tokens	512 tokens	5120 tokens	7168 tokens
FAQ	33.1	41.0	31.9	79.8	83.8	80.6
FAQ + UT	50.6	60.1	48.0	85.9	90.6	86.1

tuning. Additionally, while the LongT5 architecture can encode large context lengths up to 5120 tokens, it is not able to reach the same performance in terms of automatic evaluation scores on a sequence longer than 5120 tokens. We tested this approach on a HR use-case but this could be implemented to any use-case that has some domain-specific FAQs with context documents available.

7 FUTURE WORK

Finally, as our experiments were impacted by the common computing constraints, there are some directions we were unable to explore and could be possible future work.

Model Selection: While LongT5 seems like a reasonable choice given the landscape of models that cater to our privacy and length constraints. However, the latest LLM developments (Chiang et al., 2023; Taori et al., 2023), also make a strong claim for their use in an FAQ chatbot. Given the overwhelmingly good performance of chatGPT and GPT 4, they could be used as a baseline for all future work in this project. The latest open-source models derived from LLaMa (Touvron et al., 2023) and other foundational LLMs like MPT-7B (MosaicML, 2023) with a context window of up to 84K, seem quite promising given their more comprehensive instruction following abilities.

Evaluation Methods: We chose to evaluate the chatbot’s performance with the standard ROUGE score given its overwhelming presence in contemporary publications, along with BERTScore based on its ability to understand text contextually. Nevertheless, for a question-answering or instruct-tuning use case, there may be other approaches to performance evaluation worth considering. Furthermore, with the surge in LLM capacity comes the ability to have automatic Natural Language Generation Evaluation (Liu et al., 2023) without relying on traditional metrics such as ROUGE and BLEU (Papineni et al., 2002) which have known shortcomings in correlation with human quality judgment (Reiter and Belz, 2009). These LLM approaches don’t necessarily rely on a reference text for evaluation, dropping the need for laborious anno-

tation work by human experts. Lastly, the inclusion of human evaluation might be costly but it is the only reliable form of evaluation and should be included in the next rounds of evaluation.

Dense Passage Retrieval: Finally, we would like to explore integrating our system with a vector database⁴, replacing our Dense Passage Retrieval (DPR) module with an enterprise-ready solution. These databases utilize document embeddings as keys such that retrieval becomes powered by vector similarity. Moreover, since they leverage efficient vector libraries like Faiss (Johnson et al., 2019), this integration would simplify our system by effectively removing the DPR component and the need for its training, making it more robust and lightweight.

ACKNOWLEDGEMENTS

The work outlined in this paper is part of a bigger research project between Technical University of Munich and SAP SE under SAP@TUM Collaboration Lab. The authors would like to thank Patrick Heinze, Albert Neumueller, Darwin Wijaya from the SAP IES as well as the Domain Experts from the Human Resource department for their continued support.

REFERENCES

- Albarede, L., Mulhem, P., Goeuriot, L., LePape-Gardeux, C., Marie, S., and Chardin-Segui, T. (2022). Passage retrieval on structured documents using graph attention networks. In Hagen, M., Verberne, S., Macdonald, C., Seifert, C., Balog, K., Nørnvåg, K., and Setty, V., editors, *Advances in Information Retrieval*, Cham. Springer International Publishing.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and

⁴<https://www.pinecone.io/>

- Amodei, D. (2020). Language models are few-shot learners.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- Guo, M., Chen, Y., Li, X., Liang, X., Liu, F., Sun, J., and Zhou, M. (2021). Longt5: Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv:2112.07916*.
- Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Karpukhin, V., O’Connor, B., Stokowiec, W., Humeau, Y., Raison, M., Foster, J., Yih, W.-t., Gao, J., Le, Q., and Wolf, T. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2021). Supervised contrastive learning.
- Leite, B. and Lopes, H. (2023). Towards enriched controllability for educational question generation.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81. Association for Computational Linguistics.
- Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). G-eval: Nlg evaluation using gpt-4 with better human alignment.
- Liu, Z., Michel, P., and Liu, X. (2020). Etc: Encoding long and structured inputs in transformers.
- MosaicML (2023). Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-16.
- Nicolescu, L. and Tudorache, M. (2022). Human-computer interaction in customer service: The experience with ai chatbots—a systematic literature review. *Electronics*, 11:1579.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Phang, J., Zhao, Y., and Liu, P. (2023). Investigating efficiently extending transformers for long input summarization. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3946–3961, Singapore. Association for Computational Linguistics.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.
- Rajpurkar, P., Zhang, J., Lachlan, R., and Manning, C. D. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model. *GitHub repository*.
- Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2022). Efficient transformers: A survey.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.
- Ushio, A., Alva-Manchego, F., and Camacho-Collados, J. (2023). A practical toolkit for multilingual question and answer generation. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Toronto, Canada. Association for Computational Linguistics.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners.
- Xiong, W., Gupta, A., Toshniwal, S., Mehdad, Y., and tau Yih, W. (2022). Adapting pretrained text-to-text models for long text sequences.
- Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.
- Zaib, M., Zhang, W. E., Sheng, Q. Z., Mahmood, A., and Zhang, Y. (2021). Conversational question answering: A survey.
- Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.